TRANSISTOR - TRANSISTOR MICROLOGIC® INTEGRATED CIRCUITS

by John Nichols

INTRODUCTION

The Fairchild 9000 series TTμL is a complete line of high-level Transistor-Transistor Micrologic ® integrated circuits offering high speed, high fanout, and high noise immunity at relatively low power consumption. A typical gate propagation delay of 6 nsec is coupled with a typical power dissipation of only 11 mw per gate. Due to the high capacitive load driving capabilities, the typical delay is just 11 nsec with 150-pf load at 25°C. This high performance is obtained with a single power supply of 5 volts.

The 9000 series of the TTμL elements are available in the 1/4-by-1/4-inch flat package, or the Dual-In-Line* package. The logic configuration, pin layout, and loading constants for all circuits in the family are shown in Fig. 1.

*Fairchild patent pending.

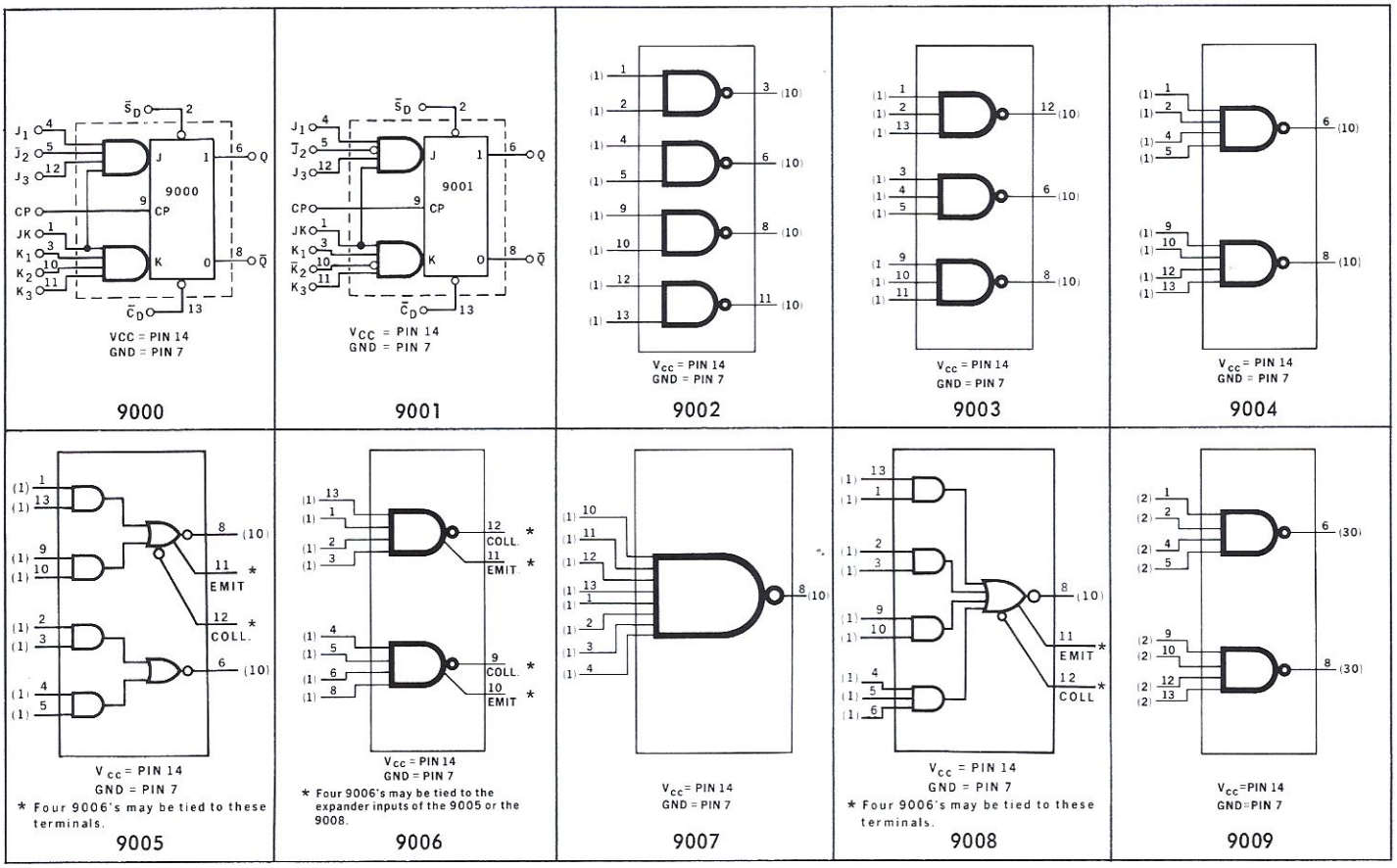


Fig. 1. Transistor-Transistor Micrologic ® IC family.

COPYRIGHT, FAIRCHILD SEMICONDUCTOR, 1967 • PRINTED IN U.S.A. 2320-486-126 30M

313 FAIRCHILD DRIVE, MOUNTAIN VIEW, CALIFORNIA, (415) 962-5011, TWX: 910-379-6435

FAIRCHILD SEMICONDUCTOR
A DIVISION OF FAIRCHILD CAMERA AND INSTRUMENT CORPORATION

Fairchild cannot assume responsibility for use of any circuitry described other than circuitry entirely embodied in a Fairchild product. No other circuit patent licenses are implied.

DESCRIPTION OF GATING ELEMENTS

The basic gating elements implement the NAND (low output only when all inputs high) or the NOR (high output if any input low) logic function. In accordance with MIL-806 B, the symbol for the basic gates may be drawn as shown in Fig. 2.

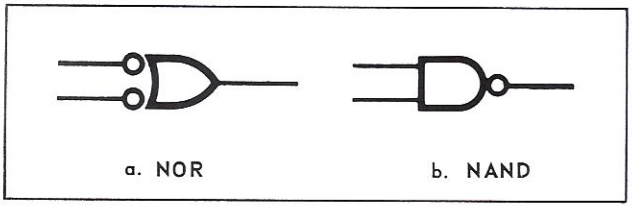


Fig. 2. Basic gate symbols.

Which symbol is chosen depends chiefly upon how the logic designer views the function being implemented. As an example, he might draw the implementation of A · B + C · D = F as shown in Fig. 3a if he chooses the high level to represent an assertion. If a low level is chosen as an assertion, Fig. 3(b), which is electrically the same as Fig. 3(a), implements the function F = (A + B) · (C + D).

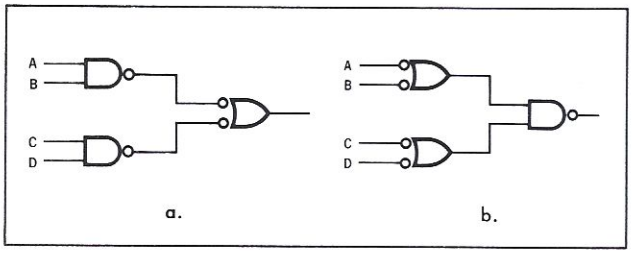


Fig. 3. Electrically equivalent logic circuits.

The basic gate is provided in four configurations: the quad two-input version (9002), the triple three-input version (9003), the dual four-input version (9004), and the single eight-input version (9007).

Supplementing the basic gates for the purpose of providing additional fanout is the TTμL 9009 dual four-input NAND/NOR power gate with a fanout of 30, as compared with fanout of 10 for the basic gates. This additional load-driving capability is obtained with only a 2-ns increase in through delay.

Additional gating capability is provided by the TTμL 9005, 9006, and 9008 elements to implement more complex logic functions with fewer packages and less cumulative through delay than could be accomplished by using the basic gates to mechanize the same functions.

The 9005 and 9008 implement the negation of a sum of two-variable products. The 9005 as shown in Fig. 1 contains two such circuits. One circuit of the two may be expanded to provide additional four variable products by connecting the expander elements. Two expanders are contained in each 9006 element. Up to four expanders may be connected to the expandable element in the 9005.

The negation of the sum of four, two-variable products may be implemented with the 9008 package. This element may be expanded with up to four 9006's. Fig. 4 shows typical connection of the 9006 expander to the 9005 and the 9008.

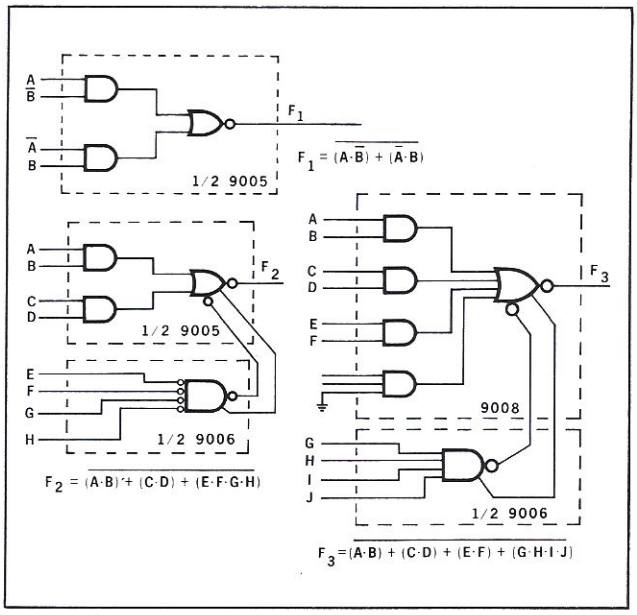


Fig. 4. Expanding using a 9006 logic element.

DESCRIPTION OF FLIP-FLOPS

The two JK, master-slave flip-flops in the TTμL family (9000 and 9001) provide four-input gates on the J and K inputs, and a direct set and direct reset that operate the flip-flop independent of the clock condition. These versatile input characteristics result in the reduction of the number of gates required with the storage elements to implement many standard functions. For some functions, no additional gates are needed. Such reduction results in a decrease in cost (fewer packages) and an increase in speed (fewer through delays).

The only difference between the 9000 and the 9001 is that the 9000 has a longer setup time (12 to 30 ns) for better clock skew immunity, and the 9001 operates faster (T_{setup} = 1 to 10 ns) and provides $J_1J_2\bar{J}_3$ and $K_1K_2\bar{K}_3$ at its synchronous input gates (Fig. 1).

The gate interconnection diagram (Fig. 5) reveals the exact logical operating characteristics of the 9000 flip-flop. Note that the 9000 is provided with an inverting buffer on the clock input so that only one load is presented to the clock. Because of this buffer, the flip-flop responds to the positive transition of the clock signal.

The 9000 flip-flop is composed of a master latch and a slave latch which operate together as follows: When the clock is high, the synchronous inputs ($J_1J_2J_3$ and $K_1K_2K_3$) are disabled and the master latch directs the condition of the slave latch. With the clock high, only the asynchronous inputs can

affect the state of the flip-flop; the flip-flop is most immune to noise from both inputs and outputs in this condition.

When the clock goes low, the slave latch is locked in its present condition, independent of the master, and the synchronous input gates are enabled. Actually, either the J or K input gate is still disabled by the JK feedback from the slave latch outputs. This cross-connection disables the J input gate when the flip-flop is in the "1" condition, and the K input gate when it is in the "zero" condition. Thus, the only change permitted due to the synchronous input gates with the clock low is for the master latch to assume a condition opposite that of the slave latch. This change will occur if all the data at the appropriate input gate are high at any time when the clock is low. When the clock subsequently goes high, the new state of the master is transferred to the slave, independent of the synchronous inputs at this time.

For proper operation of the flip-flop, the signals at the sensitive synchronous input gate must not be allowed to be unintentionally high while the clock is low if it is not desired to change the state of the flip-flop. Note that this requirement is considerably less stringent than the misleading statement, "the data must not change while the clock is low." It is clear that it makes no difference how the data change on the J inputs when the flip-flop is set, nor how the data change on the K inputs when the flip-flop is reset. Nor is there any adverse effect if in setting the flip-flop the J input signal goes high after the clock goes low but before the setup time.

In most applications, it is advantageous to use a clock with the minimum amount of time in the low level to provide the greatest noise immunity. This reduces the possibility of non-ideal operation due to data changing when the clock is low. When it is possible for the data to change when the clock is low, or when unusual operation of the flip-flop is desired, analysis of Fig. 5 will reveal the results to be expected in a particular application.

OPERATING CONSIDERATIONS

There are several operating considerations that should be taken into account in TTμL circuitry. These include loading rules, unused inputs, wiring, clock skew, and pin assignments, as will be discussed in the following paragraphs.

Loading Rules

The loading constants given in Fig. 1 are applicable over the entire specified temperature and supply voltage range. These rules provide a minimum 400-mV signal line noise immunity over the specified temperature range. At intermediate temperatures, fanout may be increased with minimal loss of noise immunity, provided that guaranteed parameters in the TTμL specification are not exceeded.

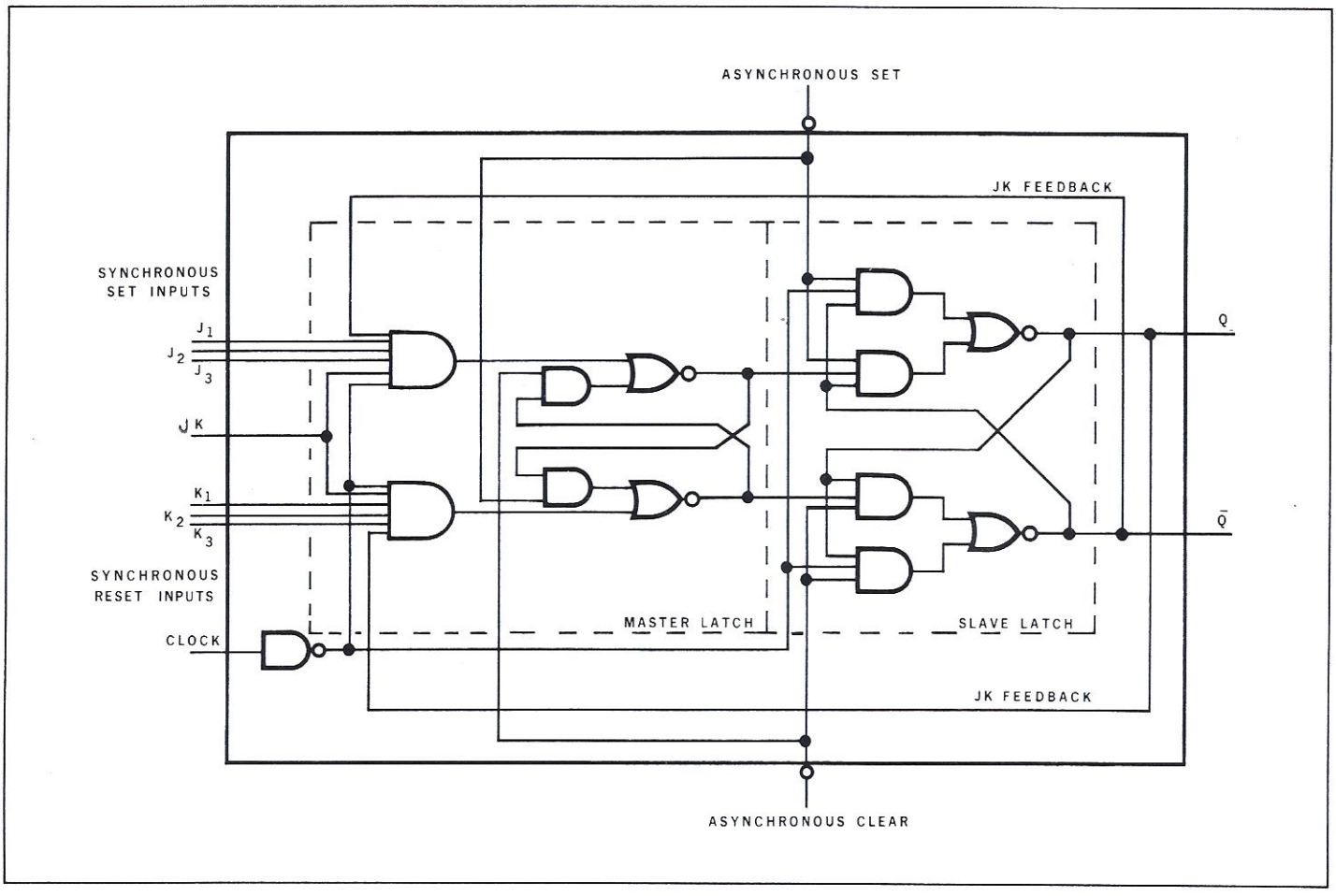


Fig. 5. Logic diagram of 9000 master-slave flip-flop.

The TTμL fanout rules apply when driving all elements of the DTμL family. The fanouts for DTμL driving TTμL over the full military temperature range are a fanout of 1 for DTμL elements having 6-K pullup resistors, a fanout of 5 for 2-K pullup resistors, and larger fanouts for buffered output devices such as the DTμL 9932, 9944 (with suitable pullup resistor), and 9950. Increased fanout for more limited temperature ranges may be extrapolated from data sheets.

Unused Inputs

Unused inputs may be left open (no connection). However, maximum operating speed is obtained with TTμL elements when all inputs are used, because of the shunt capacitance of unused inputs. The adverse effects of unused input capacitance may be minimized in one of the following ways. First, an unused input may be connected to another logic signal having sufficient fanout, provided this signal does not adversely affect the logical operation of the element. Second, the unused input may be connected to V_{CC} if the allowable range of V_{CC} (+8.0 volts) is limited to the more restricted allowable range of the inputs (+5.5 volts).

The $\overline{J}$ and $\overline{K}$ inputs on the 9001 must have a connection to ground when not used and the unused AND gates in the 9008 and 9005 elements must have at least one input grounded for proper logical operation.

Wiring

The characteristically fast rise and fall times of TTμL require certain precautions in wiring details, such as lead lengths and dress. V_{CC} and ground lines should be carefully routed and decoupled as in any high-speed system. The TTμL output circuitry is capable of driving large capacitive loads, so that low-impedance transmission lines are not required or even desirable in most cases. Inductive effects in interconnecting wiring can become even more important than capacitive effects. Capacitive loading becomes significant, however, when using the 9006 expander. The speed and noise immunity of the 9006 can be maximized by avoiding both capacitive loads and long-line lengths.

Clock Skew

High-speed logic requires consideration in areas which could be ignored with slower logic. One of these areas results from the fact that the clock signal will not occur at exactly the same time at all flip-flops due to delays along interconnecting lines. When flip-flops are not driven by the same elements, the difference in through delay causes additional time differences in the occurrence of the clock. This difference in occurrence is referred to as "clock-skew". When delays associated with the operation of the flip-flop are relatively large, misoperation due to clock skew is less likely to occur. But when fast operating flip-flops such as the 9000 and the 9001 are used, the effects of clock skew must be evaluated. Consider the example in Fig. 6. Flip-flop "B" receives its inputs through some logic delay from flip-flop "A". As an example of clock skews, flip-flops A and B are shown receiving clocks C_1 and C_2, which are logically intended to occur at the same time, but actually occur at different times.

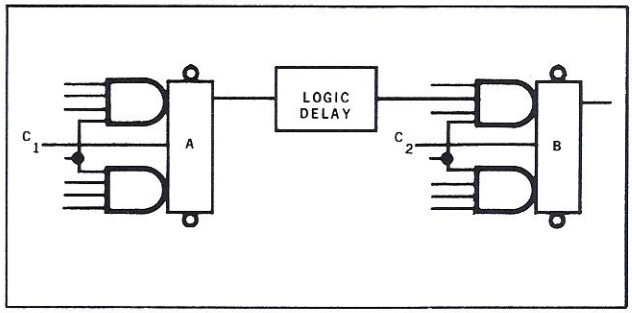


Fig. 6. Example of clock skew.

Correct operation of this circuit would be for flip-flop B to assume the state after clock C, as indicated by the state of flip-flop A prior to the clock. We assume that, prior to the clock, sufficient time has elapsed since the last clock for all logic signals to have reached their desired levels. Correct operation will occur if C_2 occurs either prior to C_1 or a short time after C_1. If the occurrence of C_2 is delayed sufficiently that the next state of flip-flop A propagates out of flip-flop A, through the logic delay and into the input of flip-flop B, then misoperation will occur in that flip-flop B will respond, not to the state of flip-flop A prior to the clock, but to its state after the clock. Correct operation will occur if

$$\text{Clock skew} \leq t_{pd} + (\text{min}) + t_{\text{release}}\ (\text{min}) + t_{\text{logic delay}}\ (\text{min}),$$

where clock skew is the maximum time that C_2 can occur after C_1, $t_{pd} + (\text{min})$ is the minimum time for the output to assume the correct state after the clock. Thus, $t_{\text{logic delay}}\ (\text{min})$ is the minimum time through the logic. And $t_{\text{release}}\ (\text{min})$ is the minimum time prior to the clock (for TTμL) that a change in input may occur and still affect the state of the flip-flop after the clock.

Operation in relation to skew is controlled through the factors in the equation above. Output delay is the only factor not easily controlled. For large logic delays between flip-flops, no skew problem will result. The 9000 flip-flop has been designed to have a large hold time in order to reduce skew problems. In those applications where the data flows in one direction; i.e., in certain shift register applications, flip-flop A does not depend on flip-flop B, and the skew might be controlled to be opposite the flow of data so that C_2 is

forced to occur before C_1, either by adding through delays or by routing of wiring.

Notice that clock repetition rate is not related to clock skew except indirectly; in that, at high clock rates, the logic delay will have been reduced and the 9001 flip-flop will have been selected for its short hold time.

Flip-Flop Pin Assignments

Pin assignments for the 9000 and 9001 flip-flops were designed to minimize or eliminate crossovers on printed circuit interconnections. Maximum advantage of the pin assignments may be obtained by first routing the power traces under elements as shown in Fig. 7; and second, by recognizing the symmetrical nature of the flip-flop (either output may be assigned the role of assertion output). To provide as much flexibility as possible, similar functions were provided on both sides of the package (since the opposite sides are separated by the power traces). As an example: J and K inputs as well as an asynchronous input were brought out to both sides and one of the Q outputs was brought out to each side. If the required Q output or asynchronous input

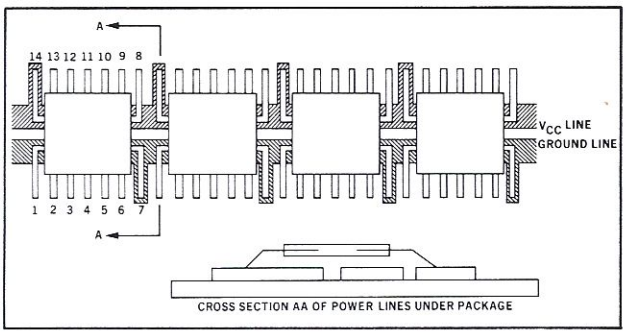
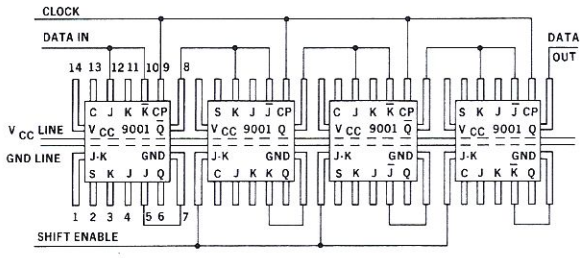


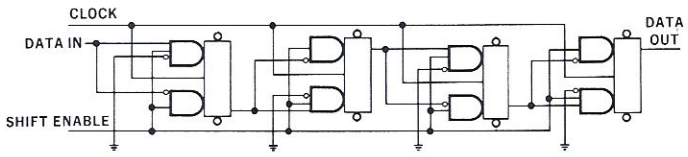
Fig. 7. Suggested power lead routing for single-sided PC board.

is not on the proper side for the interconnection requirements, this may be taken care of by using the symmetrical nature of the flip-flop. Since the clock input in many cases is connected in common to several flip-flops, this pin was assigned between the input pins and the output pin on one side.

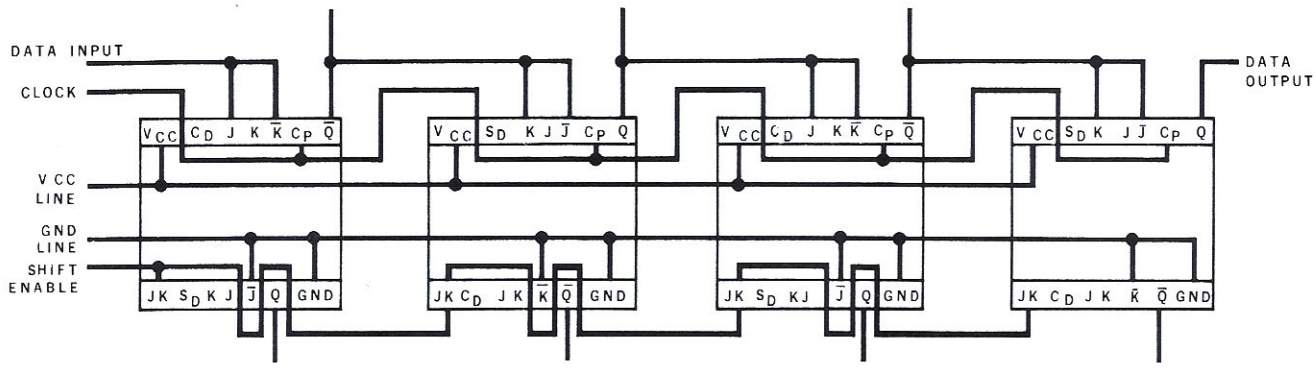
Figures 8, 9, and 10 show the trace layout required, using a single-sided board to implement three common logic circuits with 9000 and 9001 flip-flops. In Fig. 8, four bits of a



a. PC board layout.



b. Logic diagram.



c. PC board layout for 14-pin DIP.

Fig. 8. Shift register with enable input.

shift register with an enable input are shown. Notice that no crossovers are required and that the definition of the assertion output is alternated. The same interconnection advantages accrue to the 14-pin Dual-In-Line* package with the additional features that traces may be routed between pins and more traces are allowed under the package (see MIL STD-806). Fig. 8(c) shows the printed circuit layout for the

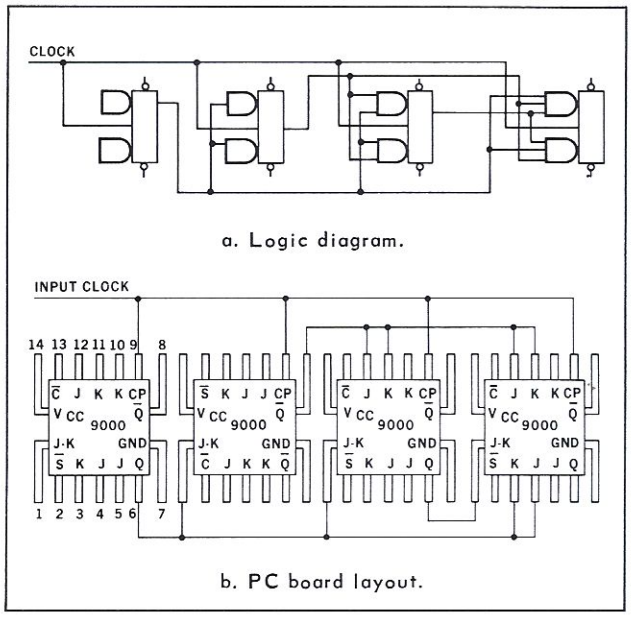


Fig. 9. Synchronous binary counter.

14-pin Dual-In-Line* Package. Fig. 9 shows a synchronous four-bit binary counter which requires no additional gates

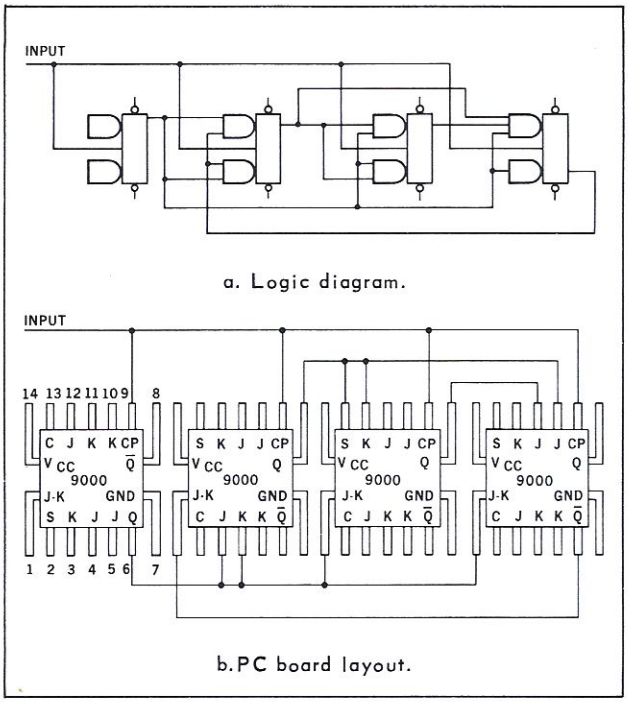


Fig. 10. Synchronous BCD (8-4-2-1) decade counter.

and only one crossover. No additional gates and only one crossover are also characteristics of the synchronous 8-4-2-1 BCD counter shown in Fig. 10.

APPLICATIONS

Gating Circuits

Logic circuits composed of inverting gates are often easier to understand, and/or construct if they are related to circuits made up of the more familiar AND, OR, and NOT functions.

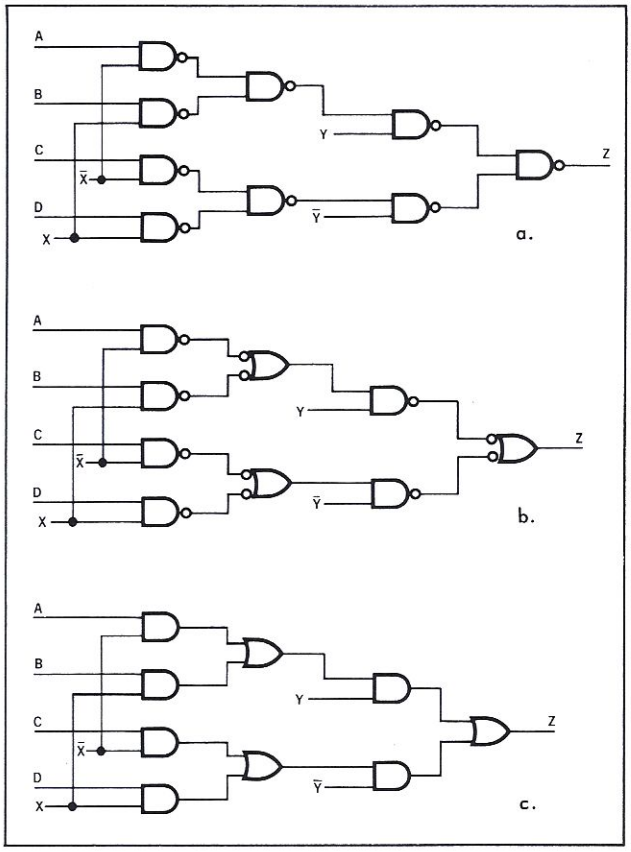


Fig. 11. Conversion from NAND to AND-OR.

Several methods may be used to perform this transformation. One method is illustrated in Fig. 11. Fig. 11(a) shows a four-logic-level connection of NAND gates. Each element in Fig. 11(a) may be represented by either the NAND symbol, as shown, or by the NOR symbol. In Fig. 11(b), the NAND representative at the alternate levels has been replaced by NOR symbols. Notice that now the circled outputs only drive circled inputs. Since the circle output indicates that the active output is the low-voltage level, and the circle input indicates that the activating input is the low-voltage level, the omission of the circles will not change the symbolic opera-

*Fairchild patent pending.

tion. Fig. 11(c) shows the circuit composed of AND-OR elements that perform the same functions as the circuits shown in Fig. 11(a) and Fig. 11(b), which are constructed of inverting gates. The methods illustrated are to replace alternate levels of the same type of converting gate by alternate levels of AND and OR elements.

The circuits shown in Fig. 11 may be used as a digital multiplexer which transfers the condition of the A, B, C, or D input line to the output line Z depending upon the state of the X and Y input lines. The function is clearer to most logic designers in the form shown in Fig. 11(b) or (c).

Another technique is illustrated by the modification to the digital multiplexer, shown in Fig. 12. This circuit has the unusual characteristic that the Z output, which may be due to the C or D inputs, is suppressed if both the A and B input lines are high. In this example, all logic paths do not contain the same number of logic levels; therefore, the change to alternate AND-OR elements results in an ambiguity. After converting some symbols from the NAND to NOR representation as before, we find one case where a circled input goes to other than a circled input. This is accounted for in the conversion to AND-OR-NOT logic by the inclusion of a phantom inverter as shown in Fig. 12(c).

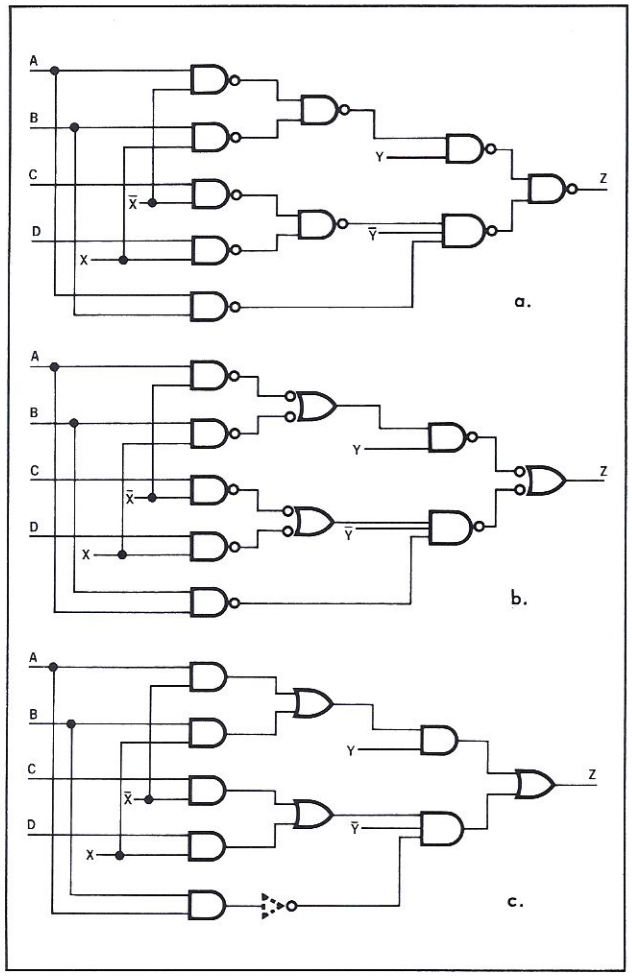


Fig. 12. Conversion from NAND to AND-OR-NOT.

The construction of logic functions using 9005's, 9008's, and 9006's can be approached by inplementing the negation of the function required, since the output is the negation of a sum of products. As an example, Fig. 13 shows the truth table, Karnaugh map, logic equation, and logic diagram for a two-bit comparator. Notice the reduced products under the negation are obtained by implementing the zero portion of the Karnaugh map.

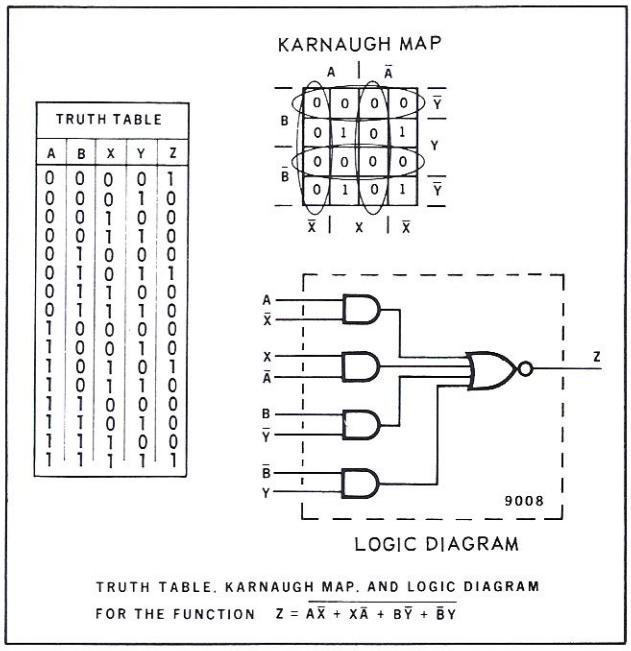

KARNAUGH MAP
TRUTH TABLE
A B X Y Z
0 0 0 0 1
0 0 0 1 0
0 0 1 0 0
0 0 1 1 0
0 1 0 0 0
0 1 0 1 1
0 1 1 0 0
0 1 1 1 0
1 0 0 0 0
1 0 0 1 0
1 0 1 0 1
1 0 1 1 0
1 1 0 0 0
1 1 0 1 0
1 1 1 0 0
1 1 1 1 1
9008
LOGIC DIAGRAM
TRUTH TABLE, KARNAUGH MAP, AND LOGIC DIAGRAM
FOR THE FUNCTION Z = AX̄ + XĀ + BȲ + B̄Y

Fig. 13. Truth table, Karnaugh map, and logic diagram.

Another example is an EXCLUSIVE-OR gate with half a 9005, shown in Fig. 14, along with the same element used as a single-bit COMPARE function, which is the negation of EXCLUSIVE-OR function.

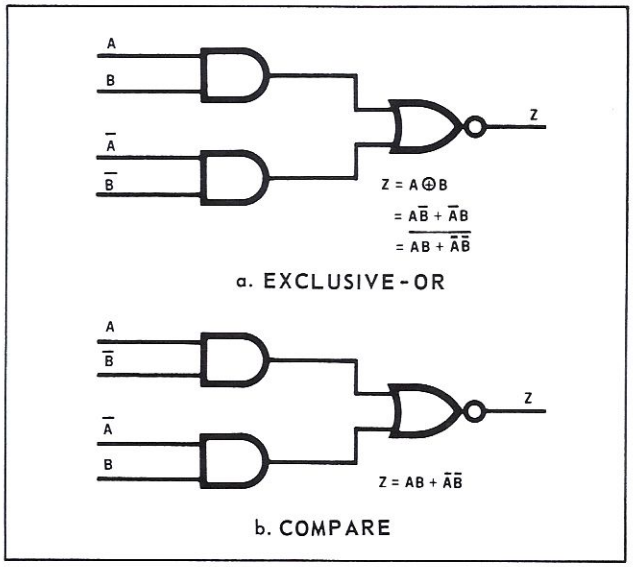


Fig. 14. EXCLUSIVE-OR and COMPARE functions.

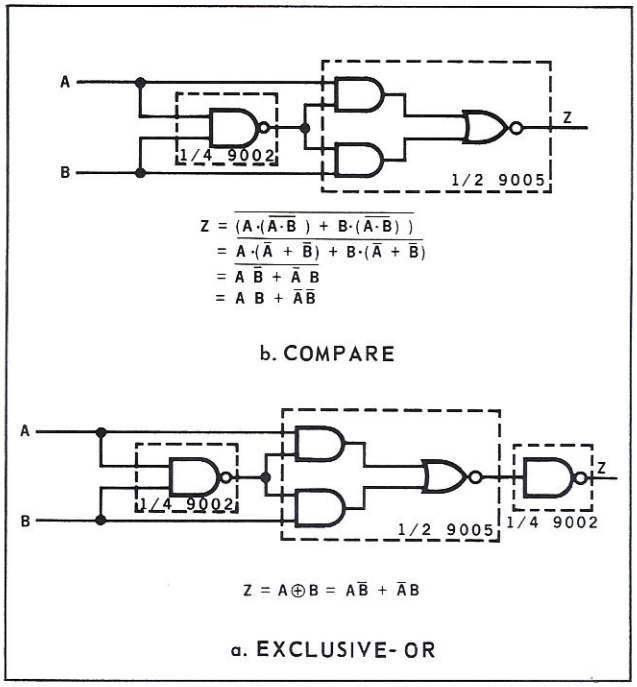


Fig. 15. EXCLUSIVE-OR and COMPARE (inverted inputs not required).

Fig. 15 shows another method for producing the COMPARE function when the negations are not available. If an inverter is added, then the EXCLUSIVE-OR function is again generated. Although requiring three propogation delays this EXCLUSIVE-OR circuit is advantageous in applications such as the full adder circuit shown in Fig. 16. The sum output is the EXCLUSIVE-OR of all the inputs; i. e., $A \oplus B \oplus C$, which may be formed in two EXCLUSIVE-OR levels, $(A \oplus B) \oplus C$. Notice that an intermediate output of the second EXCLUSIVE-OR is used to generate the carry. Only the assertion of the carry is generated or required.

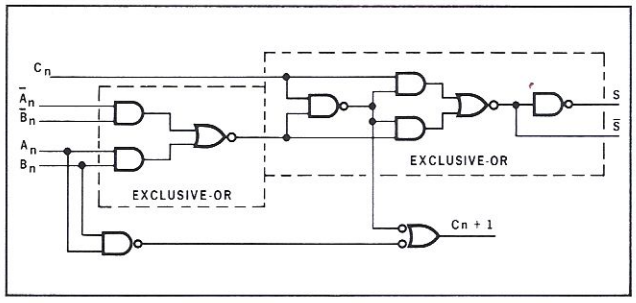


Fig. 16. Full adder using one 9005 and one 9002.

The generation or checking of parity is another example of the use of the EXCLUSIVE-OR function. Fig. 17 shows a circuit in which both versions of the EXCLUSIVE-OR gate are used for minimum hardware implementation.

Fig. 18 shows the construction of a N-bit binary to gray converter that requires (N-1) EXCLUSIVE-OR gates.

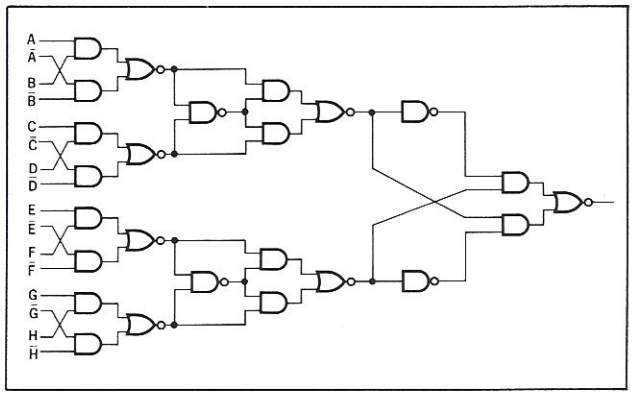


Fig. 17. Parity generator or checker.

Gate Storage Circuits

For some storage requirements, it is often less expensive to use a combination of gates instead of a standard flip-flop. An example of a simple storage circuit using 1/2 a 9002 is

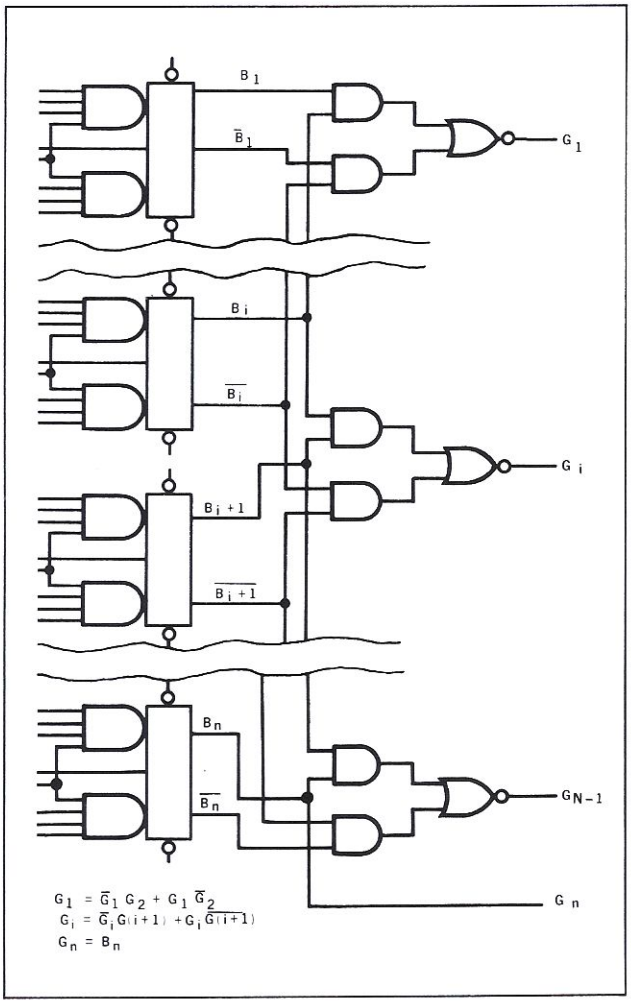


Fig. 18. N-bit binary to gray converter.

shown in Fig. 19. When both A and B inputs to this circuit are high, the output X will be high if A was low last. Except for the condition when both A and B inputs are low,

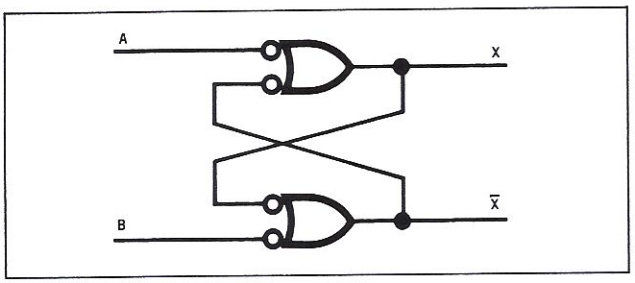


Fig. 19. Simple "latch".

the X and $\overline{X}$ outputs will be logical complements of each other. In this condition (both inputs low), both outputs will be high. This circuit, commonly referred to as a "latch", has many applications. As an example, it can be used as a "1's catcher" to receive data from a monitor tape-recorder single-shot multivibrator. For this purpose, the output of each read-head single-shot is applied to the A input and all the B inputs are connected to a common clear signal, as shown in Fig. 20.

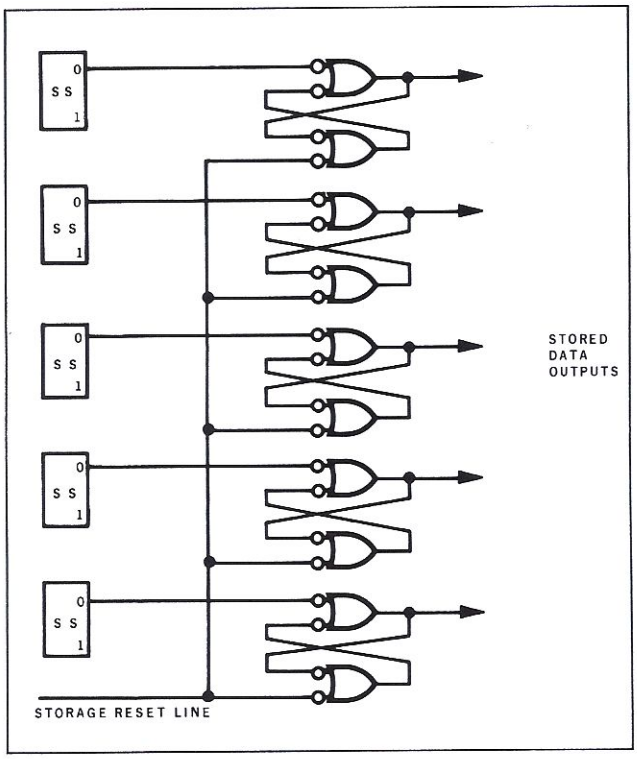


Fig. 20. "Ones catcher" storage register.

Fig. 21 gives two of the many simple storage circuits constructed of gates that may provide limited data storage capability at less cost than a flip-flop. The circuit in Fig. 21(a) has a gate added to the latch (Fig. 19) so that a transfer of "1's" may be made from a common data buss after the clear input has reset the latch. Fig. 21(b) shows the connection necessary to produce a more complex latch circuit with a single 9005. If inputs S_1 and S_2 are both high and no other conflicting inputs are present, the latch will either remain in or will transfer to the set condition (the X output high and the $\overline{X}$ output low). In the same way, inputs R_1 and R_2 will reset the latch. A low provided to the S_3 input sets the latch, and a low provided to R_3 resets the latch.

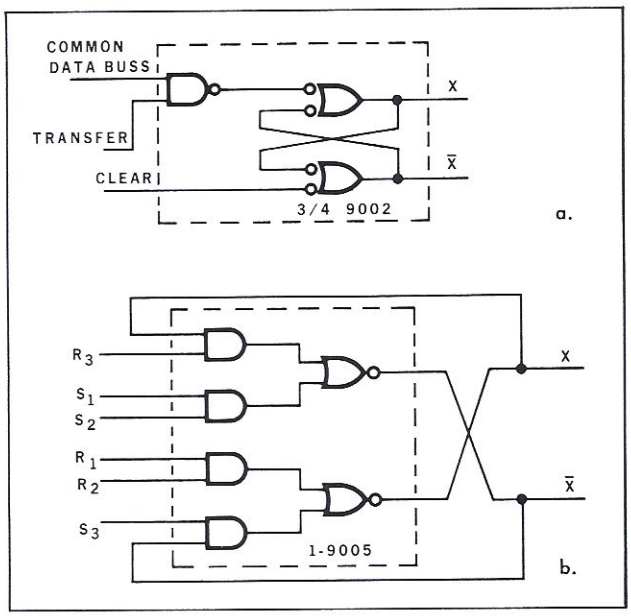


Fig. 21. Gate storage circuits.

Flip-Flop Counting Circuits

The design of counting circuits begins with a decision of the order in which the counter states shall occur, continues with the establishment of rules for the operation of each flip-flop to accomplish the selected order, and is completed by the implementation of these rules with logic circuitry.

The binary counting sequence is a common order for the states of a counter, and in this sequence the rule for each flip-flop is simple, That is, each flip-flop will change condition (toggle) when all the less significant flip-flops are in the "1" condition. A counter which uses this rule for each flip-flop is commonly called an "up counter". The same counter may also be considered a "down counter" if the definition of the "1" and the "zero" outputs of each flip-flop is reversed. This is because the rule for counting down is that a flip-flop will toggle when all the less significant flip-flops are in the zero condition.

In either case, the least significant flip-flop toggles on every count input. Fig. 9 shows four 9000 flip-flops connected together to form a binary counter. The logic required to produce the binary count is provided at the inputs to the 9000 flip-flop, so that no additional gating is required. Fig. 10 shows the interconnection necessary to produce a counter that goes to the zero state (0000) after the binary nine state (1001), commonly known as a decade counter. Either of the

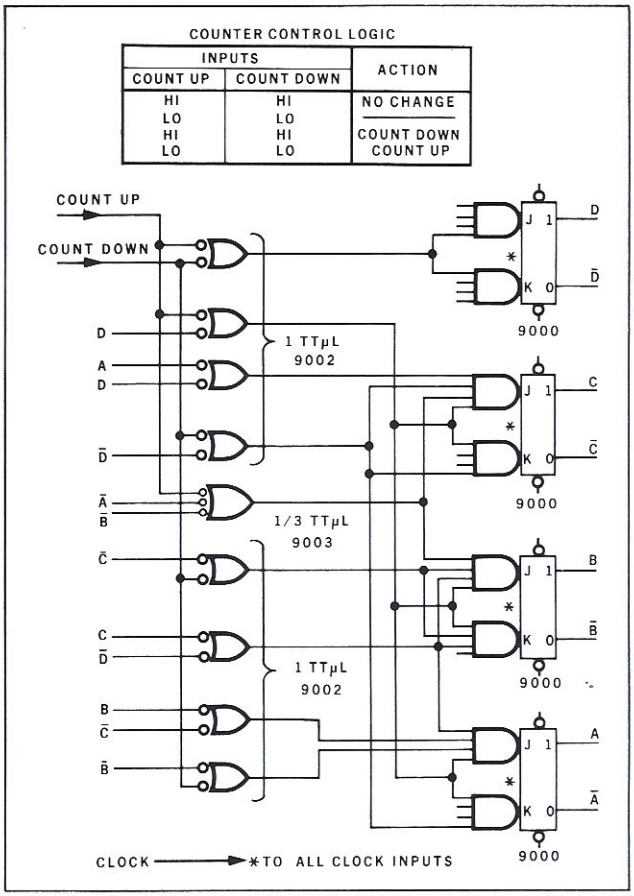


Fig. 22. Synchronous up/down BCD counter.

counters (Fig. 9 or Fig. 10) may be considered as a single counting stage. Any number of such stages may be connected together to produce larger counters (without requiring additional gates) by connecting the zero output of the most significant bit of one stage to the count (clock) input of the next more significant stage.

A synchronous up/down BCD counter using four 9000 flip-flops and nine gates (6-1/3 TTμL packages) is shown in Fig. 22. This counter has the feature that when both control inputs are high, the counter does not change in response to clock signals.

SUMMARY

The intent of this application bulletin has been to supplement the specification sheets for the Transistor-Transistor Micrologic ® integrated circuit family (TTμL) and to provide detailed information concerning the logical operation of various elements in the family. Methods for implementing logic functions and illustrations of various logic circuits have been included to assist logic designers in the use of the TTμL family.